In the Claims:

1.   (Previously Presented): A system for passing messages from a first application to a second application in a distributed application server comprising:

a message modulator at a first entity for modulating a message, each modulated message having a flexible message header and a plurality of typed container modules;

a message receiver at a second entity for demodulating said typed container modules to regenerate said message; and

wherein said message modulator operates in an edit mode for editing and modulating a message, and in a storage mode for storing a message, wherein in said edit mode, each typed container module, except those containing user data, is prefixed with an attachment unit which comprises pointers to point to the next typed container module and the previous typed container module, and wherein in said storage mode the attachment unit is removed from the typed container modules.

2.   (Previously Presented):   The system of claim 1 wherein said typed container modules include a typed container header portion defining the type and length of the typed container module header; and,

a typed container body portion containing a portion of information.

3.   (Original): The system of claim 1 wherein one of said typed container modules includes a user data portion.

- 2 -

4.   (Original): The system of claim 1 wherein said message receiver demodulates only a subset of said typed container modules to create said message.

5.   (Original): The system of claim 1 wherein said typed container modules are linked to each other by pointers.

6.   (Original): The system of claim 1 wherein said typed container modules are linked to said flexible message header by pointers.

7.   (Original): The system of claim 1 wherein the message header part of the flexible message header comprises an attachment unit for linking to said typed container modules.

8.   (Canceled)

9.   (Currently amended): The system of claim 8- 1, wherein in said storage mode, the message is stored in one continuous memory space and all of the typed container modules stored without their pointer set.

10.   (Previously Presented):   A method of messaging between applications in a distributed application system, comprising the steps of:

- 3 -

modulating a message, each modulated message having a flexible message header and a plurality of typed container modules;

demodulating said typed container modules to create a message; and

wherein said message modulator operates in an edit mode for editing and modulating a message, and in a storage mode for storing a message, wherein in said edit mode, each typed container module, except those containing user data, is prefixed with an attachment unit which comprises pointers to point to the next typed container module and the previous typed container module, and wherein in said storage mode the attachment unit is removed from the typed container modules.

11.   (Previously Presented):   The method of claim 10 wherein said typed container modules include:

a typed container header portion defining the type and length of the typed container module header; and,

a typed container body portion containing a portion of information.

12.   (Original):   The method of claim 10 wherein one of said typed container modules includes a user data portion.

13.   (Original):   The method of claim 10 wherein said message receiver demodulates only a subset of said typed container modules to create said message.

- 4 -

14.   (Original): The method of claim 10 wherein said typed container modules are linked to each other by pointers.

15.   (Original): The method of claim 10 wherein said typed container modules are linked to said flexible message header by pointers.

16.   (Original): The method of claim 10 wherein the message header part of the flexible message header comprises an attachment unit for linking to said typed container modules.

17.   (Canceled)

18.   (Currently amended): The method of claim 4̶7̶ 10 , wherein in said storage mode, the message is stored in one continuous memory space and all of the typed container modules stored without their pointer set.

19.   (Withdrawn):   A method for messaging between applications in a distributed application system, comprising the steps of:

        generating a message at a first application, together with message header information and body information;

        segmenting the header information and body information into container modules;

- 5 -

creating a flexible message header;

attaching to each container module an attachment unit containing pointers linking the container module to the flexible message header and to each successive container module;

sending the message as a series of typed container modules to a second application; and,

selecting at said second application certain of the container modules and reconstructing the message.

20.   (Withdrawn): The method of claim 19 wherein the message includes a user data portion.

21.   (Withdrawn): The method of claim 20 further comprising the step of:

placing said user data portion into a user data module and linking said user data module to said flexible message header.

- 6 -